

# Introduction to AVR<sup>®</sup>

By BiPOM Electronics, Inc.

Revision 1.02

© 2011 BiPOM Electronics, Inc. All rights reserved.

All trademark names in this document are the property of their respective owners.

# AVR<sup>®</sup> History

- The AVR<sup>®</sup> architecture was conceived by two students at the Norwegian Institute of Technology.
- The original AVR<sup>®</sup> was known as  $\mu$ RISC (Micro RISC).
- Among the first of the AVR<sup>®</sup> line was the AT90S8515, which in a 40-pin DIP package has the same pinout as an 8051 microcontroller.
- The creators of the AVR<sup>®</sup> give no definitive answer as to what the term "AVR" stands for.

# AVR<sup>®</sup> Features

- Some 8-bit, some 32-bit
- TinyAVR, megaAVR, XMEGA, FPSLIC
- Harvard Architecture for 8-bit devices: Separate code and data space
- Flash, EEPROM and SRAM are all on a single chip, eliminating the need for external memory.
- All code executed by the AVR<sup>®</sup> core must reside in the on-chip flash.
- Most instructions take just one or two clock cycles.
- The AVR family of processors were designed with the efficient execution of compiled C code in mind.

# Why AVR<sup>®</sup> ?

- The AVR<sup>®</sup> instruction set is more powerful than PIC or 8051.
- The AVR<sup>®</sup> runs instructions very fast (can execute 1 instruction in 1 machine clock cycle)
- AVR<sup>®</sup> is a good choice for industrial projects.

# Frequently Asked Questions

Can AVR<sup>®</sup> run an OS?

- Yes, AVR32 can run Linux core 2.6.XX with BusyBox

What programming languages are for programming AVR<sup>®</sup> microcontroller?

- There are many different languages but most commonly used are C and BASCOM BASIC.

Does BiPOM offer AVR<sup>®</sup> design services ?

– Yes, we are a certified ATMEL consultant



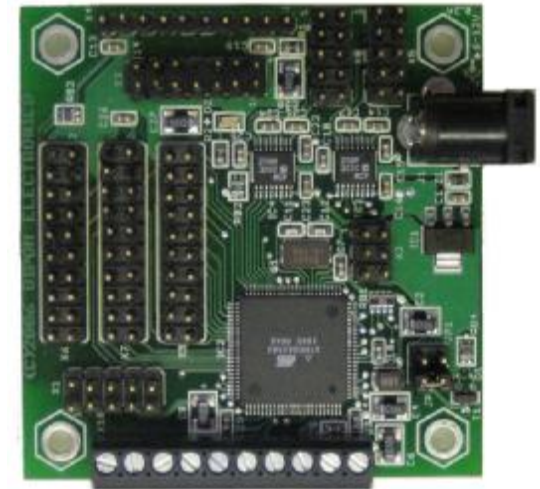
# BiPOM AVR® Support



MINI-MAX/AVR-AU  
General Purpose  
plus USB support  
based on  
AT90USB162-16MU



MINI-MAX/AVR-BU  
General Purpose,  
USB support,  
Analog inputs  
based on  
AT90USB647



MINI-MAX/AVR-C  
General Purpose  
based on  
ATMEGA2560-16



# BiPOM AVR® Support



MOTOR-2  
Motor Controller  
based on  
ATMEGA168V-10



MINI-MAX/AVR  
Set I & Set II



MicroTRAK/AVR-C Starter  
and  
MicroTRAK/AVR-C Complete  
Development/Training Kits



# AVR<sup>®</sup> Development Tools

- Flowcode for AVR
- Flowcode Support Package for AVR
- BASCOM-AVR BASIC Compiler
- ATMEL AVR Studio
- WinAVR AVR GCC Compiler for Windows

# Flowcode for AVR<sup>®</sup>

Flowcode 1

File Edit View Panel View Macro Run Chip Window Help

Objects - Common - Inputs - Outputs - Coins - Wireless - Peripheral - Mechanics - Misc

Main

Flowchart:

- BEGIN
- Decision:  $e > 10?$
- If Yes: Disable TMFO
- If No: Calculation ( $e = 10$ )
- Switch e
- Case 1: Disable PORTB, Disable TMFO
- Case 2: LED ON LED ON, LED ON LED ON
- Case 3: While  $e > 10$ , KeyPad() GetByte(), loop
- Case 4: Disable TMFO, C Code: Enter C co. Alternatively...
- END

Chip: atmega161p

Pin	Function
PORTB0/D0/D0	PB0
PORTB1/D1/D1	PB1
PORTB2/D2/D2	PB2
PORTB3/D3/D3	PB3
PORTB4/D4/D4	PB4
PORTB5/D5/D5	PB5
PORTB6/D6/D6	PB6
PORTB7/D7/D7	PB7
PORTC0/D8/D8	PC0
PORTC1/D9/D9	PC1
PORTC2/D10/D10	PC2
PORTC3/D11/D11	PC3
PORTC4/D12/D12	PC4
PORTC5/D13/D13	PC5
PORTC6/D14/D14	PC6
PORTC7/D15/D15	PC7
PORTD0/D16/D16	PD0
PORTD1/D17/D17	PD1
PORTD2/D18/D18	PD2
PORTD3/D19/D19	PD3
PORTD4/D20/D20	PD4
PORTD5/D21/D21	PD5
PORTD6/D22/D22	PD6
PORTD7/D23/D23	PD7
PORTD8/D24/D24	PD8
PORTD9/D25/D25	PD9
PORTD10/D26/D26	PD10
PORTD11/D27/D27	PD11
PORTD12/D28/D28	PD12
PORTD13/D29/D29	PD13
PORTD14/D30/D30	PD14
PORTD15/D31/D31	PD15

Variables:

Variable	Type	Value
e	BYTE	0 (0x0)

Properties:

Panel

Property Value

Appearance

CapColor Panel

Size 181, 610

Position 47, 726

Misc

Name Panel

Background

Color #ccc9d8

Image

File

Node Center

Grid

Size 8, 8

Show False

Snap True

Analogue Window

ADC0: \_\_\_\_\_

ADC1: \_\_\_\_\_

ADC2: \_\_\_\_\_

ADC3: \_\_\_\_\_

ADC4: \_\_\_\_\_

ADC5: \_\_\_\_\_

ADC6: \_\_\_\_\_

ADC7: \_\_\_\_\_

Panel

Call Stack:

Macro Call:

Main

For Help, press F1

Current zoom = 75%

CALL 3844, 3281

# Flowcode for AVR®

- One of the world's most advanced graphical programming languages for microcontrollers
- Allows creating complex electronic systems in minutes without prior experience.
- Supports a large set of AVR® microcontrollers
- Provides a large set of ready to use components like LED's, LCD's, EEPROM, Serial, I/O, etc.
- Easy to use graphical interface
- Includes Simulator, including support for components

# BASCOM-AVR

The screenshot displays the BASCOM-AVR IDE interface. The main window shows a code editor with the following content:

```
Sub  
'name : function bas  
'copyright : (c) 1995-2005, MCS Electronics  
'purpose : demonstrates FUNCTION  
'micro : Mega48  
'suited for demo : yes  
'commercial addon needed : no  
'  
$regfile = "m48def.dat" ' we use  
$crystal = 8000000  
$baud = 19200  
  
$hwstack = 32  
$swstack = 16  
$framesize = 24  
  
'A user function must be declare before it can be used.  
'A function must return a type  
Declare Function Myfunction(byval I As Integer , S As String) As  
'The byval paramter will pass the parameter by value so the original  
'will not be changed by the function  
  
Dim K As Integer  
Dim Z As String * 10  
Dim T As Integer  
'assign the values  
K = 5  
Z = "123"  
  
T = Myfunction(k , Z)  
Print T  
End  
  
Function Myfunction(byval I As Integer , S As String) As Integer  
'you can use local variables :  
Local T As Integer  
  
P = I  
  
'because I is passed by value  
'variables named k  
I = 10  
  
P = Val(s) + I  
  
'finally assign result
```

The right-hand pane shows the 'Chip PinOut' configuration window. It includes a 'Package' dropdown set to 'DIP28', a 'Search' field, and buttons for 'Chip Search' and 'Clear Pin I/L'. Below this is a visual representation of a DIP28 chip.

Overlaid on the bottom right is the 'Sample Electronics AVR programmer' window. It shows the following details:

- Chip: AT90S1200
- Manufacturer: Unknown
- Flash ROM: 2KB
- EEPROM: 128
- Size: 4 bars (full)
- Programmed: 0

The programmer window also displays a memory dump table:

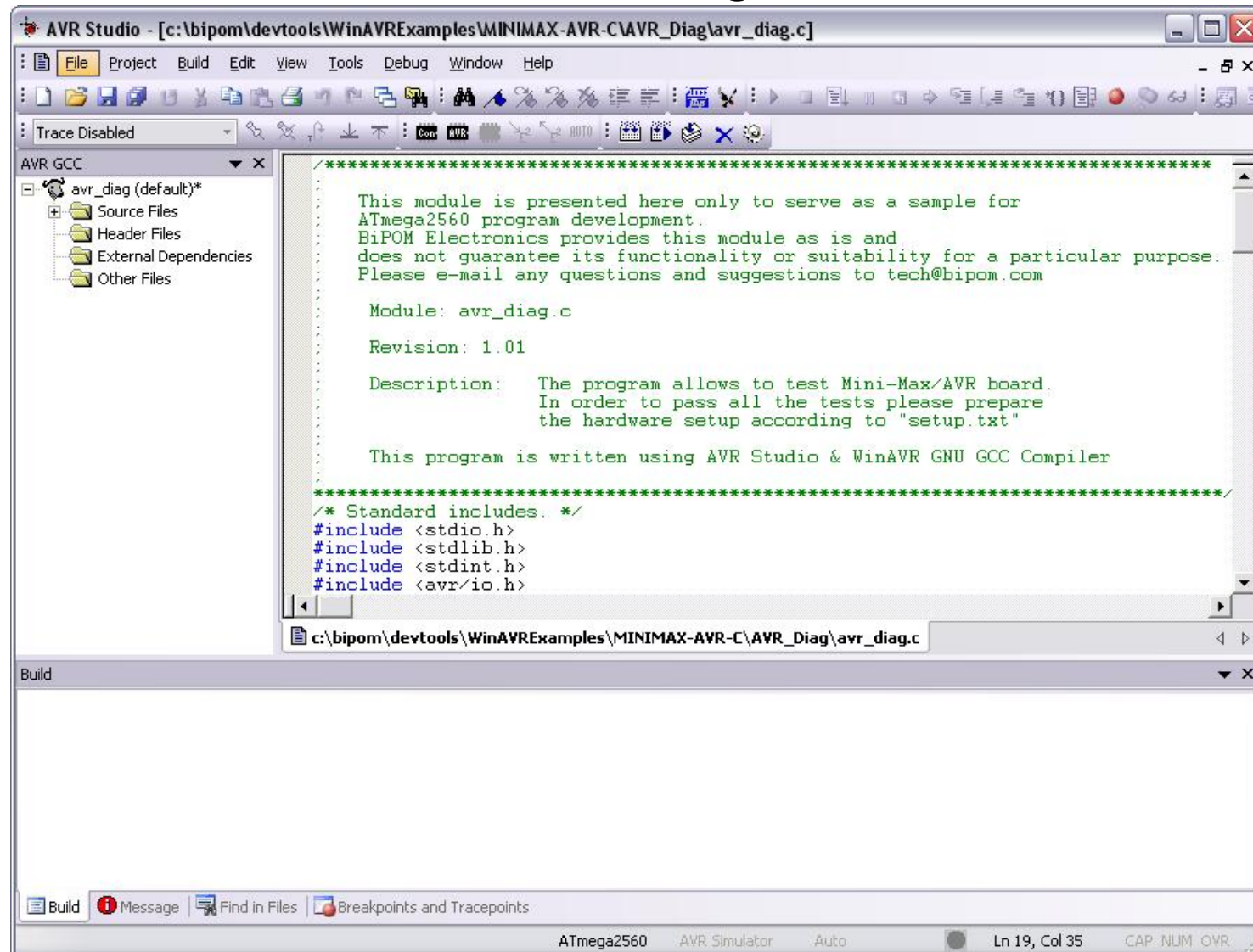
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
000	CD	19	55	18	95	18	95	18	95	18	95	18	95	13	95	18	A.....
010	95	18	55	18	95	18	95	18	95	18	95	18	95	13	95	18	.....
020	95	18	55	18	95	18	95	18	95	18	95	18	95	13	95	18	.....
030	95	10	C5	10	CF	0F	0F	0D	CC	C3	CD	C0	2C	4C	C0	02	.....nJiKaAaNa
040	BF	8E	E0	22	EC	F2	2E	5F	EF	E3	E0	F1	E0	AD	E0	B1	TFaTer_noaca at

At the bottom of the programmer window, it indicates '720 bytes read' and shows the file path 'FUNCTION.BIN'.

# BASCOM-AVR

- Windows-based Integrated Development Environment
- Editor, Project Manager
- Syntax Coloring
- Powerful BASCOM BASIC Compiler
- Simulator and Debugger
- Supports a large variety of Programmers:  
ISP, AVR-ISP, USB-ISP, STK500
- Includes many example projects

# AVR Studio by ATMEL



# AVR Studio by ATMEL

- Free
- Windows-based Integrated Development Environment
- Editor, Project Manager
- Syntax Coloring
- Integrated Debugger
- Integrated Assembler
- Integrated Simulator
- Integrates with GCC compiler plug-in
- Support for all Atmel tools that support the 8-bit AVR® architecture

# Debugging Support



**ATJTAGICE2**  
JTAG Debugging



**ATAVRONEKIT**  
On-Chip Debugging

BiPOM is an authorized reseller for ATMEL  
Development Tools and Debuggers



# Programming/Downloading Support

- USBISP: BiPOM
- ChipProg-ISP : Phyton
- ChipProg-40 : Phyton
- ChipProg-48 : Phyton
- ChipProg-G4 : Phyton
- ATAVRDRAGON : ATMEL
- AVRISP mkII: ATMEL
- Pololu USB AVR Programmer

